

LABYRINTH

GAME

1. **Content…………………………………………………………1**
2. **Abstract……………………………………………………….. 2**
3. **Description of the project..……………………. ……………. 3**
4. **Principe of algorithm………………………………………… 4**
5. **Structure……………………………………………………….6**
6. **Annexe………………………………………………………….7**

**ABSTRACT**

The main goal of our project is to create a labyrinth game with different mazes depending on the difficulties. In order to do so, we’ll create classes which will initialize the maze and characters.

* **Maze:**

Player will be able to choose a difficulty which will define the size of the maze. For example for the difficulty=5 we’ll have a maze of 45x80 and if the difficulty=1, player will came across of a maze 9x16.

* **Character:**

This class aims to create different types of characters which will be put in the maze. We’ll use the top view of characters and each time the player makes a turn the picture used in the program will change meaning that if the character is looking at right when the player presses left the face of the character will turn to left.

Firstly, in the beginning of the game player will be asked to choose a difficulty, the character that he/she wants to play with and the level of help he/she wants (1 being the least and the 3 being the most). Help will be in a relation with the number of the case that the player will be able to see around his/her character. To begin with we’ll create 4 basic classes;

* + **Maze**
  + **Player**
  + **LabyrinthDisplay**
  + **DistanceFilter**
  + **FirstFrame**

Further in the project (when we have a working program), we can add other types of subclasses in order to make our project more appealing. For example we can increase the variety of the character or change the picture of the maze.

The class maze will initialize the game area. Inside the game the green/red parts are going to be considered as the walls and the bright white parts as the roads inside the maze. Player class creates the characters and the changes direction of the character depending on the pressed button. DistanceFilter will be the class which creates “Help” which defines the blocks that the player can see around his/hers character depending on the choosen level of help. The main reason for DistanceFilter to be a separate class is to decrease the complexity because the complexity is already O(2n+m with n and m being the length and width of the maze respectively. LabyrinthDisplay will be class which visualizes the maze. Finally, FirstFrame class is will create the page where the player can choose the difficulty, help and the character.

**DESCRIPTION OF PROJECT**

Our main objective is to create a game which will contain different mazes with respect to different levels of difficulties. The difficulty (from 1 to 5) and the help (from 1 to 3) will be chosen by the player before the game starts. The higher the difficulty is the bigger the maze will be and help determines the radius that the player will be able to see around his/her character with 1 being the smallest radius and 3 being the biggest radius.

First of all, we create the class **Maze** which will generate the maze. Depending on the chosen difficulty by the player the number of white blocks and the complexity of the answer will change. The solution of the maze will be written as a method in this class.

Then, we create the class **Player** which will initialize the characters that will be used in the game. Moreover this class includes the method which changes the image used in the game with respect to the pressed button. Meaning that if a character is facing right and the player presses “4” then the character will face left. In order to realise this we used for different images for every character available in the game. Also we used the top view of images.

Next, the class **DisplayFilter** is where we define the method which allows the player to see the blocks around his/hers character. Furthermore, the player will choose the radius of visibility that he/she wants before the game stars.

Later on, the class **LabyrinthDisplay** is where we will be displaying the labyrinth with all of its components such as the bushes, the character inside the maze, the road, background etc. Also, this class will check rather a player can make a move or not. Meaning that if the character of the player is blocked by a wall at right when player presses “6” this class will check if there is a white block where the player wants to move if yes his/hers character will move if not it won’t make any move.

Finally, the class **FirstFrame** will initialize the page where the player can choose his/her character, the difficulty that he/she wants to play with and the help that he/she wants.

Of course since we’ll be dealing with IHM we used some images for characters and maze. In order to avoid any copyright violation every image that we used are designed and upload by ourselves.

**PRINCIPE OF ALGORITHM**